

What is Claimed is:

1. A method of tuning a database query, comprising:

selecting a database query;

parsing the selected database query to determine relationships between portions of the selected database query;

selecting an optimization mode from a plurality of available optimization modes;

tuning the selected database query by modifying at least one portion of the selected database query based on the determined relationships and the selected optimization mode; and

displaying the modified database query.

2. A method as recited in claim 1, wherein the parsing determines tokens within the database query, tokens being words separated by delimiters.

3. A method as recited in claim 1, wherein the plurality of available optimization modes include Cost based and Rule based modes.

4. A method as recited in claim 3, wherein the Cost based modes include First_Rows mode and All_Rows mode.

5. A method as recited in claim 1, further comprising determining a cost associated with using the tuned database query.

6. A method as recited in claim 5, further comprising comparing a cost associated with using the selected database query to the cost associated with using the tuned database query.

7. A method as recited in claim 1, further comprising parsing the selected database query to determine whether the database query includes at least one subquery joined by at least one of a

NOT EXISTS, a NOT IN and an ALL clause.

8. A method as recited in claim 7, further comprising prompting a user to select preferences to be used during tuning based on whether the database query includes at least one of a NOT EXISTS, a NOT IN and an ALL clause.

9. A method as recited in claim 8, wherein the preferences include rewrite preferences for enabling a user to select at least one of a conversion of NOT EXISTS operators to a NOT IN operator and conversion of the selected database query to an outer join.

10. A method as recited in claim 8, wherein the preferences include rewrite preferences for enabling a user to select to convert subqueries joined by an ALL operator to a join or outer-join.

11. A method as recited in claim 8, wherein the preferences include rewrite preferences for enabling a user to select whether to use at least one of a NOT EXISTS operator and an outer-join to convert subqueries joined by a NOT In operator.

12. A computer storage medium including computer executable code for tuning a database query, comprising:

computer executable code for allowing a user to select a database query;

computer executable code for parsing the selected database query to determine relationships between portions of the selected database query;

computer executable code for allowing a user to select an optimization mode from a plurality of available optimization modes;

computer executable code for tuning the selected database query by modifying at least one portion of the selected database query based on the determined relationships and the selected optimization mode; and

computer executable code for displaying the modified database query.

13. A computer storage medium as recited in claim 12, wherein the parsing determines tokens withing the database query, tokens being words separated by delimiters.
14. A computer storage medium as recited in claim 12, wherein the plurality of available optimization modes include Cost based and Rule based modes.
15. A computer storage medium as recited in claim 14, wherein the Cost based modes include First_Rows mode and All_Rows mode.
16. A computer storage medium as recited in claim 12, further comprising code for determining a cost associated with using the tuned database query.
17. A computer storage medium as recited in claim 16, further comprising code for comparing a cost associated with using the selected database query to the cost associated with using the tuned database query.
18. A computer storage medium as recited in claim 12, further comprising code for parsing the selected database query to determine whether the database query includes at least one subquery joined by at least one of a NOT EXISTS, a NOT IN and an ALL clause.
19. A computer storage medium as recited in claim 18, further comprising code for prompting a user to select preferences to be used during tuning based on whether the database query includes at least one of a NOT EXISTS, a NOT IN and an ALL clause.
20. A computer storage medium as recited in claim 19, wherein the preferences include rewrite preferences for enabling a user to select at least one of a conversion of NOT EXISTS operators to a NOT IN operator and conversion of the selected database query to an outer join.
21. A computer storage medium as recited in claim 19, wherein the preferences include

rewrite preferences for enabling a user to select to convert subqueries joined by an ALL operator to a join or outer-join.

22. A computer storage medium as recited in claim 19, wherein the preferences include rewrite preferences for enabling a user to select whether to use at least one of a NOT EXISTS operator and an outer-join to convert subqueries joined by a NOT In operator.

23. A programmed computer system for tuning a database query, comprising:

a display for displaying at least one database queries to a user;

a user input allowing the user to select a database query from among the displayed database queries and an optimization mode from a plurality of available optimization modes; and

a processor for parsing the selected database query to determine relationships between portions of the selected database query, and for tuning the selected database query by modifying at least one portion of the selected database query based on the determined relationships and the selected optimization mode, the modified database query being displayed to the user via the display.

24. A system as recited in claim 23, wherein the parsing determines tokens withing the database query, tokens being words separated by delimiters.

25. A system as recited in claim 23, wherein the plurality of available optimization modes include Cost based and Rule based modes.

26. A system as recited in claim 25, wherein the Cost based modes include First_Rows mode and All_Rows mode.

27. A system as recited in claim 23, wherein the processor determines a cost associated with using the tuned database query.

28. A system as recited in claim 27, wherein the processor compares a cost associated with using the selected database query to the cost associated with using the tuned database query.
29. A system as recited in claim 23, wherein the processor parses the selected database query to determine whether the database query includes at least one subquery joined by at least one of a NOT EXISTS, a NOT IN and an ALL clause.
30. A system as recited in claim 29, wherein the processor prompts the user to select preferences to be used during tuning based on whether the database query includes at least one of a NOT EXISTS, a NOT IN and an ALL clause.
31. A system as recited in claim 30, wherein the preferences include rewrite preferences for enabling a user to select at least one of a conversion of NOT EXISTS operators to a NOT IN operator and conversion of the selected database query to an outer join.
32. A system as recited in claim 30, wherein the preferences include rewrite preferences for enabling a user to select to convert subqueries joined by an ALL operator to a join or outer-join.
33. A system as recited in claim 30, wherein the preferences include rewrite preferences for enabling a user to select whether to use at least one of a NOT EXISTS operator and an outer-join to convert subqueries joined by a NOT In operator.